# EAGLE

and the

# High Level Architecture

**Sponsor:**

**OSD PA&E**

**U.S. Army TRADOC Analysis Center**

**U.S. Army National Simulation Center**

**Jack Ogren**

**Feb 1997**

MITRE

# Contexts
# Eagle & the High Level Architecture

0 **Eagle Design Summary**

0 **Summary of Effort & Design**

0 **Technical Approach**

0 **Class Structures**

0 **Interactions**

0 **Event Synchronization**

0 **RTI Functionality Exercised**

0 **Testing - Results**

0 **Summary**

# Contexts
# Eagle & the High Level Architecture

- ❹ **Eagle Design Summary**
- 0 **Summary of Effort & Design**
- 0 **Technical Approach**
- 0 **Class Structures**
- 0 **Interactions**
- 0 **Event Synchronization**
- 0 **RTI Functionality Exercised**
- 0 **Testing - Results**
- 0 **Summary**

**What is Eagle anyway ??**

# Eagle Design -- Summary

0 Used by TRAC as a combat development analysis tool to study corps and division level force effectiveness issues. (Not a Training Simulation)

0 Characteristics

- Corps & below level simulation
- Resolution to Battalion or Company (Entity level w/BDSD)
- Deterministic (Stochastic w/BDSD)
- Hybrid event Structure
    - = Attrition/Ground Movement/Detection/C2 are Time-stepped
    - = Air/ADA interactions are Event-driven
- Integrates Artificial Intelligence methods and conventional combat modeling algorithms
    - Object-Oriented
    - Embedded AI Systems (expert systems)
    - Symbolic decision making
    - Lisp/KEE Programming environment
- Command and Control modeled explicitly.
    - Units execute orders and pass information based on a Battlefield Management language.

# Eagle Design -- Summary

0 Functional Representation

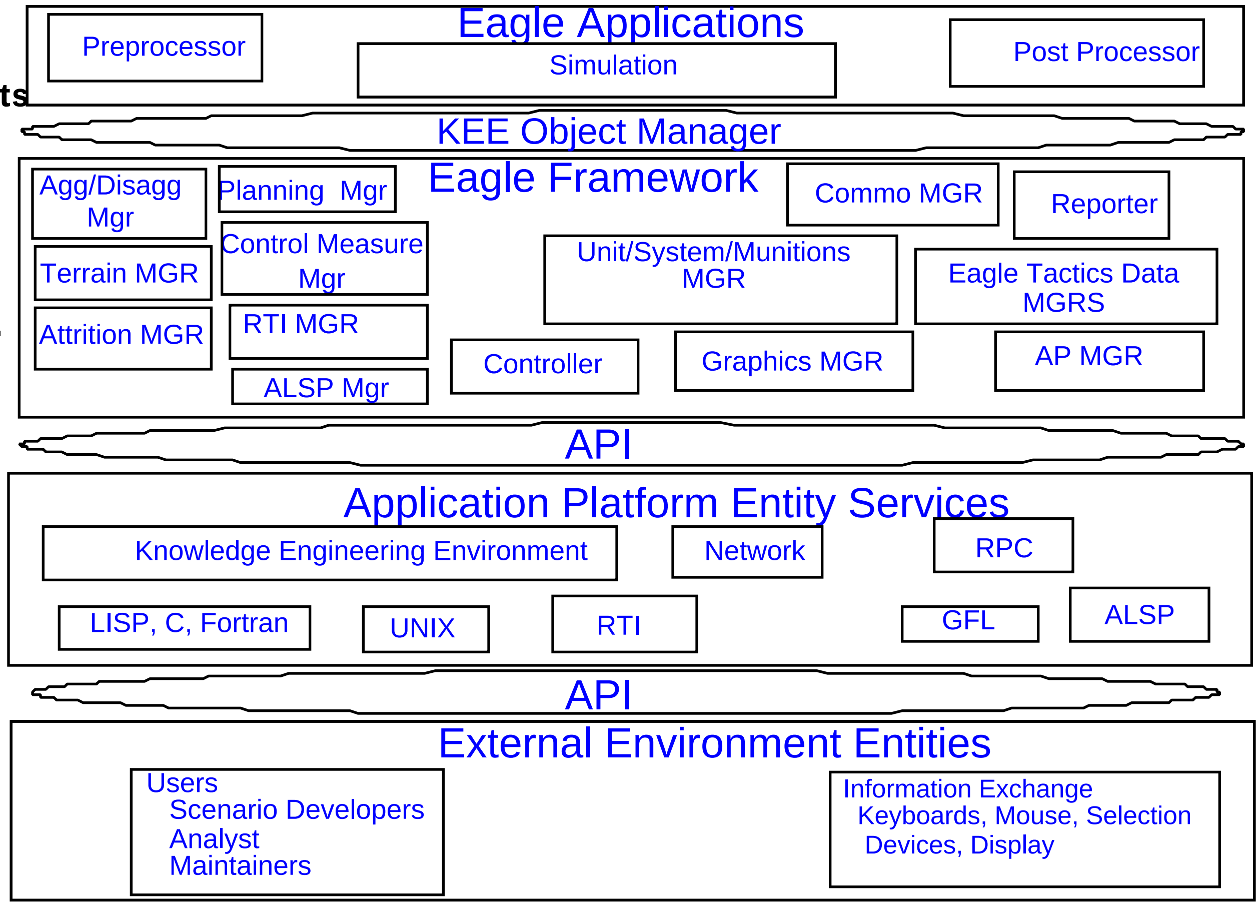
- Command & Control Headquarters (Corps through Battalion)
- Ground Maneuver Units (Company, Battalion)
- Fire Support Units (Battalion, Battery, Tube, Radar section)
- Air Maneuver Units (Battalion, Company, Section)
  - = Attack and Scout Helicopter Units
- Air Defense Units (Battalions, Battery, Sections)
- Intelligence (Sections)
  - = Air & Ground Acquisition Elements (RPV, radar, JSTARS, RECCE)
- Engineers (Battalion, Team)
  - = Mobility, countermobility, Survivability
- Fixed Wing (Flights)
  - = CAS, BAI, SEAD

# Eagle Architecture

Applications are loaded individually or combined
Simulation = Combat Units

Eagle Framework are services provided to applications.
Responsible for maintaining ground truth.

Runs on HP and Sun workstations.



Eagle Applications

Preprocessor

Simulation

Post Processor

KEE Object Manager

Eagle Framework

Agg/Disagg Mgr

Planning Mgr

Commo MGR

Reporter

Terrain MGR

Control Measure Mgr

Unit/System/Munitions MGR

Eagle Tactics Data MGRS

Attrition MGR

RTI MGR

ALSP Mgr

Controller

Graphics MGR

AP MGR

API

Application Platform Entity Services

Knowledge Engineering Environment

Network

RPC

LISP, C, Fortran

UNIX

RTI

GFL

ALSP

API

External Environment Entities

Users
Scenario Developers
Analyst
Maintainers

Information Exchange
Keyboards, Mouse, Selection
Devices, Display

# Contexts
# Eagle & the High Level Architecture

0 **Eagle Design Summary**

❹ **Summary of Effort & Design**

0 **Technical Approach**

0 **Class Structures**

0 **Interactions**

0 **Event Synchronization**

0 **RTI Functionality Exercised**

0 **Testing - Results**

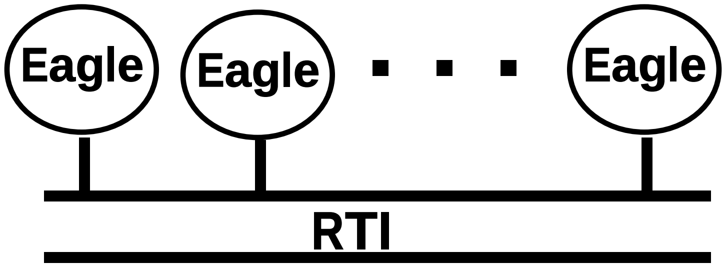
0 **Summary**

**How is Eagle playing in this effort??**

# Eagle & the HLA
# Summary of Effort to Date

## Eagle Early Analysis Experiment

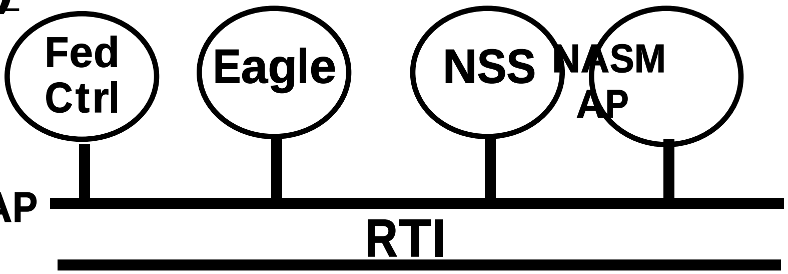
Distributed Units -
Eagle Combat units interact
using the RTI.



## Joint Training Federation (JTFp)

Distributed Functionality-
Army Combat units - Eagle
Navy Combat units - NSS
Air Force Combat units - NASM AP

Federation Controller



Design facilitate: Running as Standalone
as Distributed Units
as Distributed Functionality
as Distributed Units & Functionality (not tested)

# Distributed Eagle Goal & Basic Design

- 0 **GOAL: Decrease computation load on single processor while minimizing interactions and maintaining the same temporal, tactical, organizational and spacial consistency that currently exists on a single processor.**
  - = **Maintain current validated models.**
- 0 **DESIGN: Combat units are distributed among multiple Eagle simulations on the network.**
  - = **Each simulation maintains its own set of core services**

    **terrain, terrain evaluation, attrition, tactics DB ... etc**
  - = **Each simulation reflects all units not owned by the simulation.**
  - = **The distributed design maintains consistency between:**

    **Services on simulations (Terrain conditions).**

    **Combat units and their reflected representations.**

# Functionally Distributed Eagle Goal & Basic Design

0 **GOAL: Allow externally generated Objects (combat or environmental) to interact with Eagle generated combat units while maintaining the same or better temporal, tactical, organizational and spacial consistency that currently exists on a single processor.**

0 **DESIGN:**

- **Eagle provides all ground combat functionality for the federation. All ground combat units are reflected.**
- **Functionality within Eagle is replaced by functionality provided by the confederation members.**

  =**Eagle replaces normal fixed wing operations by**

  **Subscribing to the federation air objects (classes)**

  **Publishing & subscribing to interactions between the ground & air players (air to ground / ground to air)**

# Contexts
# Eagle & the High Level Architecture

0 **Eagle Design Summary**

0 **Summary of Effort & Design**

❹ **Technical Approach**

0 **Class Structures**

0 **Interactions**

0 **Event Synchronization**

0 **RTI Functionality Exercised**

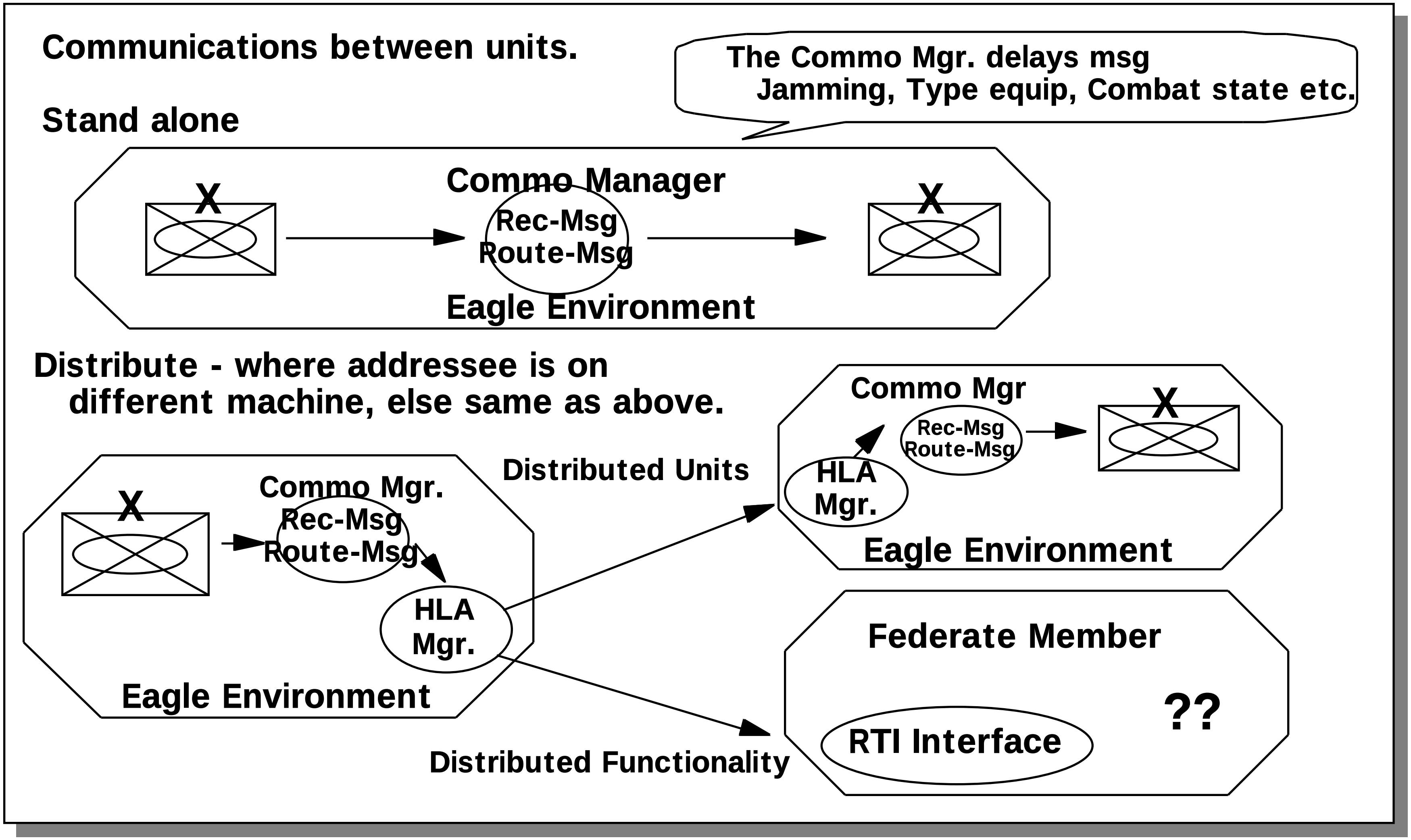
0 **Testing - Results**

0 **Summary**

{ **Within Eagle**
**and**
**Between Eagle & the RTI**

**Leverage Eagle's**
**OOD**
**&**
**Architecture**

# Technical Approach within Eagle - Example



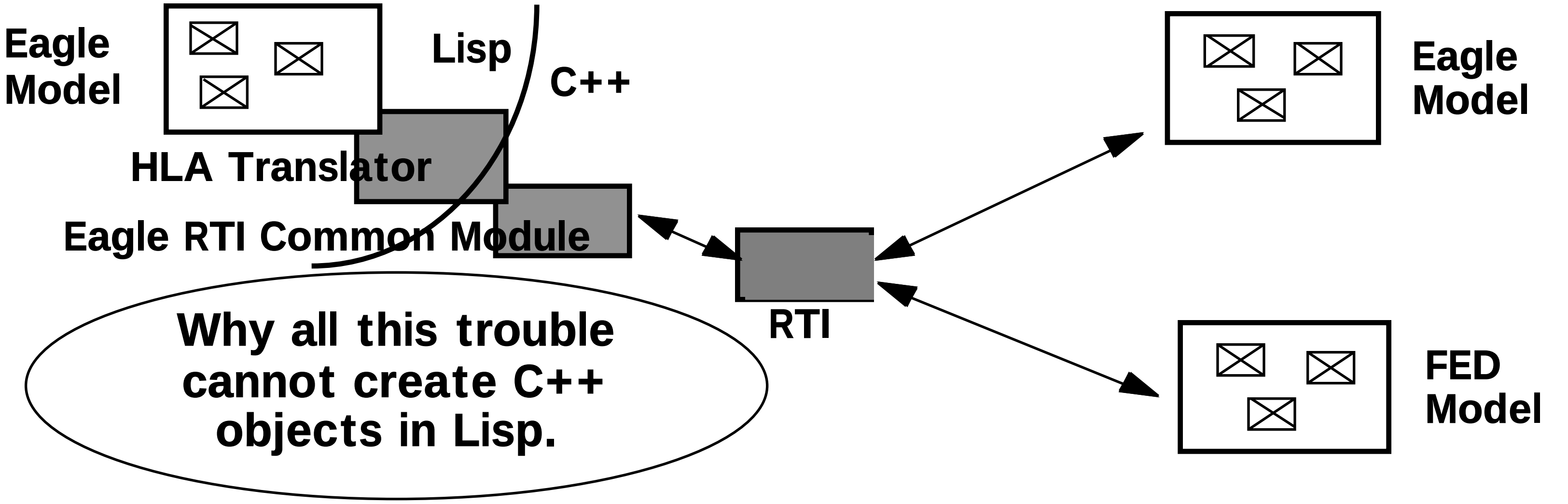
# Technical Approach between Eagle & the RTI

0 A new Eagle Service (HLA Translator) is provided within the Eagle framework to provide the interface between Eagle objects & the RTI.

- Very little was modified within the actual Eagle code. Most of the work was focused on the translator (11,000 lines of code).

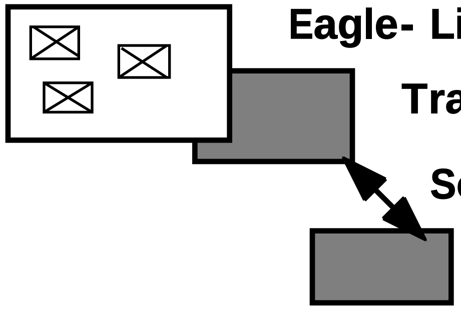
0 Each Eagle simulation HLA Translator Service (lisp) communicates with the RTI through an Eagle RTI Common module (C++). (app 4,000 lines of code)

0 The Eagle RTI Common module provides the interface with the RTI -(RTI ambassador & Federate Ambassador).



Why all this trouble cannot create C++ objects in Lisp.

# Technical Approach - External to Eagle



**Eagle- Lisp**

**Translator - Lisp/C**

**Socket Connection**

**Eagle RTI - Interface C++**

Translator determines when to send info to the socket & when to listen to the socket

**Translator - Lisp/C**

**Out the socket --> (**hla_talk “TIME_ADVANCE_REQUEST^1.34”)

**Listen to socket (Blocked ?) -->** (hla_listen)

**Eagle RTI - Interface C++** Loop - RTI event or socket Calls -Do **SOCKET CALL**

**Receive socket Stream:** Tokenize, execute Case Statements which --

**Calls rti_Ambassador -** time_advance_request(the_time, IT_X)

**Eagle RTI - Interface C++** Loop - RTI Events or socket Calls -Do RTI **EVENT**

**fed_amb method is called:** Fed_ambassador::time_advance_granted(Fed_time)

Outputs to socket “time_advance_granted 1.34”

Calls to Fed_amb are buffered on Socket when Eagle is not Listening.

# Contexts
# Eagle & the High Level Architecture

0 **Eagle Design Summary**

0 **Summary of Effort & Design**

0 **Technical Approach**

❹ **Class Structures**

0 **Interactions**

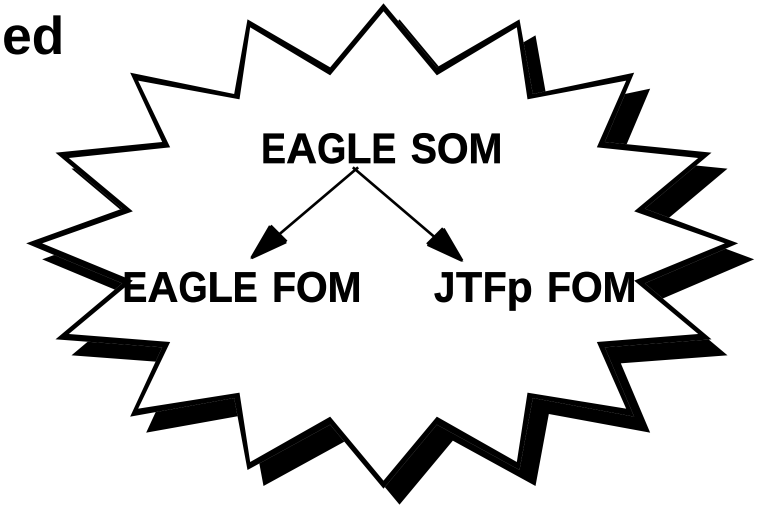
0 **Event Synchronization**

0 **RTI Functionality Exercised**

0 **Testing - Results**

0 **Summary**

**Software Object Model**
**and**
**Federation Object Model**

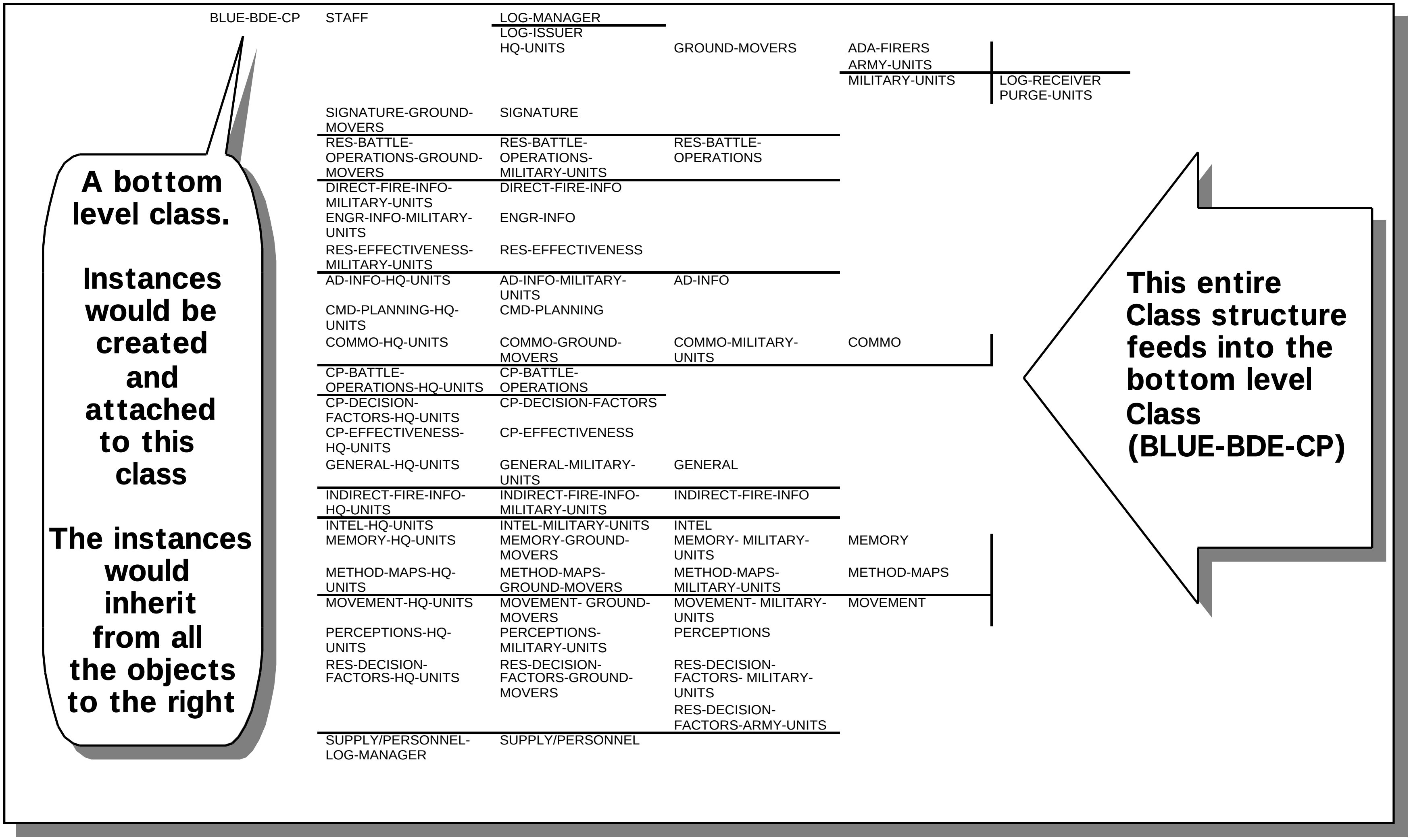


EAGLE SOM

EAGLE FOM

JTFp FOM

# Eagle Class Structure
# Normal structure in model (one example)



| | | | | | |
|---|---|---|---|---|---|
| BLUE-BDE-CP | STAFF | LOG-MANAGER | | | |
| | | LOG-ISSUER | | | |
| | | HQ-UNITS | GROUND-MOVERS | ADA-FIRERS | |
| | | | | ARMY-UNITS | |
| | | | | MILITARY-UNITS | LOG-RECEIVER |
| | | | | | PURGE-UNITS |
| | SIGNATURE-GROUND-MOVERS | SIGNATURE | | | |
| | RES-BATTLE-OPERATIONS-GROUND-MOVERS | RES-BATTLE-OPERATIONS-MILITARY-UNITS | RES-BATTLE-OPERATIONS | | |
| | DIRECT-FIRE-INFO-MILITARY-UNITS | DIRECT-FIRE-INFO | | | |
| | ENGR-INFO-MILITARY-UNITS | ENGR-INFO | | | |
| | RES-EFFECTIVENESS-MILITARY-UNITS | RES-EFFECTIVENESS | | | |
| | AD-INFO-HQ-UNITS | AD-INFO-MILITARY-UNITS | AD-INFO | | |
| | CMD-PLANNING-HQ-UNITS | CMD-PLANNING | | | |
| | COMMO-HQ-UNITS | COMMO-GROUND-MOVERS | COMMO-MILITARY-UNITS | COMMO | |
| | CP-BATTLE-OPERATIONS-HQ-UNITS | CP-BATTLE-OPERATIONS | | | |
| | CP-DECISION-FACTORS-HQ-UNITS | CP-DECISION-FACTORS | | | |
| | CP-EFFECTIVENESS-HQ-UNITS | CP-EFFECTIVENESS | | | |
| | GENERAL-HQ-UNITS | GENERAL-MILITARY-UNITS | GENERAL | | |
| | INDIRECT-FIRE-INFO-HQ-UNITS | INDIRECT-FIRE-INFO-MILITARY-UNITS | INDIRECT-FIRE-INFO | | |
| | INTEL-HQ-UNITS | INTEL-MILITARY-UNITS | INTEL | | |
| | MEMORY-HQ-UNITS | MEMORY-GROUND-MOVERS | MEMORY- MILITARY-UNITS | MEMORY | |
| | METHOD-MAPS-HQ-UNITS | METHOD-MAPS-GROUND-MOVERS | METHOD-MAPS-MILITARY-UNITS | METHOD-MAPS | |
| | MOVEMENT-HQ-UNITS | MOVEMENT- GROUND-MOVERS | MOVEMENT- MILITARY-UNITS | MOVEMENT | |
| | PERCEPTIONS-HQ-UNITS | PERCEPTIONS-MILITARY-UNITS | PERCEPTIONS | | |
| | RES-DECISION-FACTORS-HQ-UNITS | RES-DECISION-FACTORS-GROUND-MOVERS | RES-DECISION-FACTORS- MILITARY-UNITS | | |
| | | | RES-DECISION-FACTORS-ARMY-UNITS | | |
| | SUPPLY/PERSONNEL-LOG-MANAGER | SUPPLY/PERSONNEL | | | |

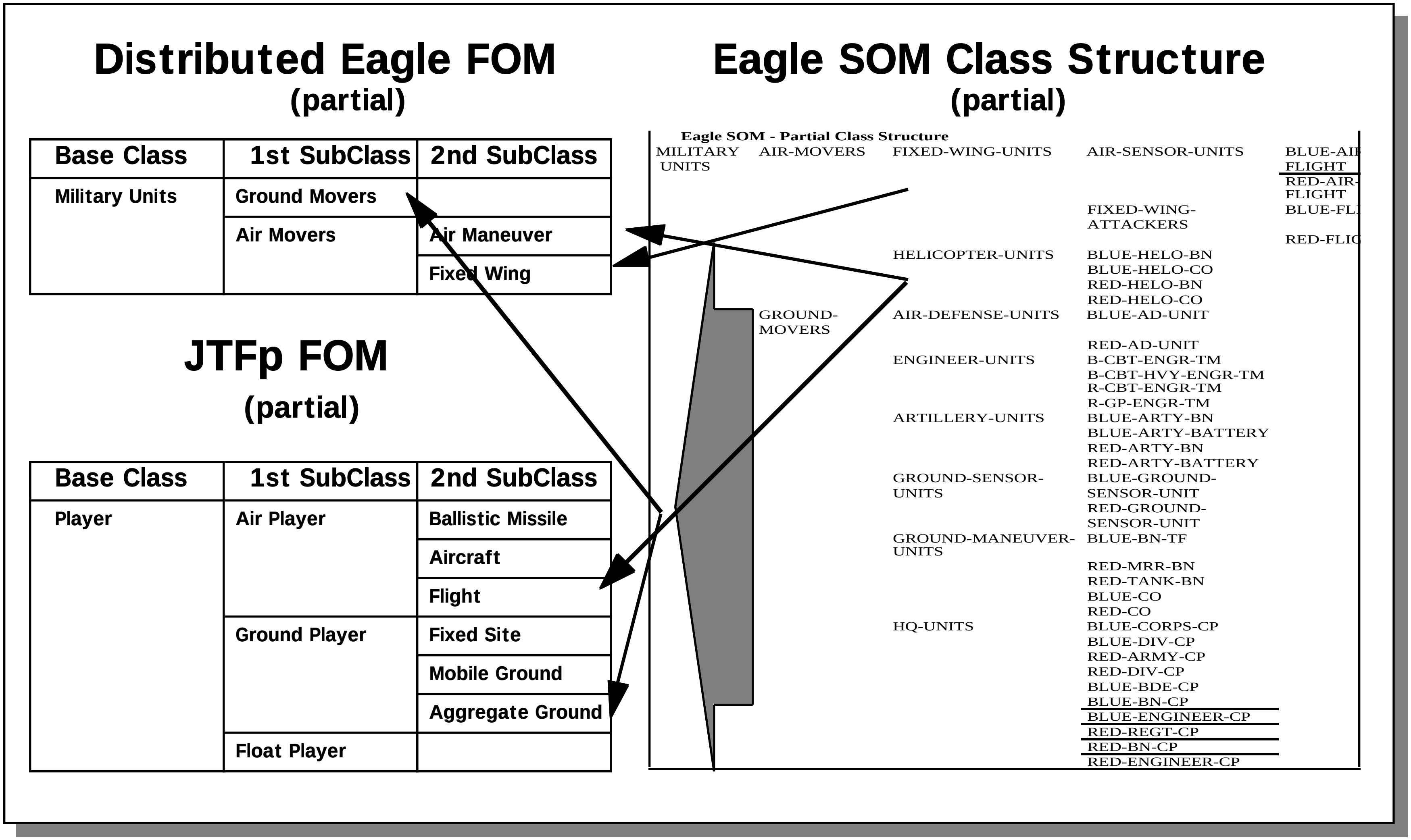
A bottom level class.

Instances would be created and attached to this class

The instances would inherit from all the objects to the right

This entire Class structure feeds into the bottom level Class (BLUE-BDE-CP)

# Eagle Class Structure ❹ Eagle SOM ❹
## Distributed Eagle FOM && JTFp FOM



### Distributed Eagle FOM
(partial)

| Base Class | 1st SubClass | 2nd SubClass |
|---|---|---|
| Military Units | Ground Movers | |
| | Air Movers | Air Maneuver |
| | | Fixed Wing |

### JTFp FOM
(partial)

| Base Class | 1st SubClass | 2nd SubClass |
|---|---|---|
| Player | Air Player | Ballistic Missile |
| | | Aircraft |
| | | Flight |
| | Ground Player | Fixed Site |
| | | Mobile Ground |
| | | Aggregate Ground |
| | Float Player | |

### Eagle SOM Class Structure
(partial)

**Eagle SOM - Partial Class Structure**

| | | | | |
|---|---|---|---|---|
| MILITARY UNITS | AIR-MOVERS | FIXED-WING-UNITS | AIR-SENSOR-UNITS | BLUE-AI<br>FLIGHT<br>RED-AIR-<br>FLIGHT |
| | | | FIXED-WING-ATTACKERS | BLUE-FL<br>RED-FLI |
| | | HELICOPTER-UNITS | BLUE-HELO-BN<br>BLUE-HELO-CO<br>RED-HELO-BN<br>RED-HELO-CO | |
| | GROUND-MOVERS | AIR-DEFENSE-UNITS | BLUE-AD-UNIT<br>RED-AD-UNIT | |
| | | ENGINEER-UNITS | B-CBT-ENGR-TM<br>B-CBT-HVY-ENGR-TM<br>R-CBT-ENGR-TM<br>R-GP-ENGR-TM | |
| | | ARTILLERY-UNITS | BLUE-ARTY-BN<br>BLUE-ARTY-BATTERY<br>RED-ARTY-BN<br>RED-ARTY-BATTERY | |
| | | GROUND-SENSOR-UNITS | BLUE-GROUND-SENSOR-UNIT<br>RED-GROUND-SENSOR-UNIT | |
| | | GROUND-MANEUVER-UNITS | BLUE-BN-TF<br>RED-MRR-BN<br>RED-TANK-BN<br>BLUE-CO<br>RED-CO | |
| | | HQ-UNITS | BLUE-CORPS-CP<br>BLUE-DIV-CP<br>RED-ARMY-CP<br>RED-DIV-CP<br>BLUE-BDE-CP<br>BLUE-BN-CP<br>BLUE-ENGINEER-CP<br>RED-REGT-CP<br>RED-BN-CP<br>RED-ENGINEER-CP | |

# Class Structure

## Attributes used to define Reflected Units

### Distributed Eagle FOM

(Partial - Ground Mover)

| OBJECT | ATTRIBUTE |
|---|---|
| MILITARY-UNITS | Battlefield Operating System |
| | Higher Hdq's name |
| | Depth of unit (meters) |
| | Direction of movement of the unit |
| | Disaggregated boolean |
| | Echelon |
| | Percent Effective |
| | Effectiveness State |
| | Frontage of Unit (meters) |
| | Latitude |
| | Longitude |
| | Name |
| | Operational Activity |
| | Orientation of weapons |
| | Phantom boolean |
| | Purged boolean |
| | Quantity of Sytems on-hand by type |
| | Route segment |
| | Side |
| | Size |
| | Systems requirements code |
| | System configuration |
| | Task |
| | Unit Type |
| GROUND-MOVERS | Air Defense Controller |
| | Air Defense Network boolean |
| | Air Defense Status |
| | Assignment |
| | Command Assignment |
| | Command Unit boolean |
| | Indirect Fire Systems on-hand |
| | Number of Indirect Fire Units |
| | Prioritized Enemy List |
| | Signature - counter battery |
| | Signature - counter mortar |
| | Signature - combat |
| | Signature - communications |
| | Previously detected boolean |
| | Signature - IR |
| | Signature - Moving Target Radar |
| | Signature - photo |
| | Net speed from last time step |
| | System weight factor |

### JTFp FOM

(Partial - Aggregate Ground Player)

| OBJECT | ATTRIBUTE |
|---|---|
| Player | entity name |
| | federate id |
| | affliliation |
| | motion type |
| | voice nets |
| | jtids nets |
| | trap tre |
| | comander type |
| Aggregate Entity | radar cross section |
| | radar detectable |
| | elint detectable |
| | comint detectable |
| | ir detectable |
| | photoint detectable |
| | air to air engageable |
| | surf to air engageable |
| | air to surf engageable |
| | surf to surf engageable |
| | composition |
| Dead Reckoned Players | time at last cse change |
| | lat at last cse change |
| | lng at last cse change |
| | alt at last cse change |
| | cse at last cse change |
| | hspd at last cse change |
| | vspd at last cse change |
| Aggregate Ground Player | depth |
| | front |
| | orientation |

Number of Attributes to define a typical ground combat unit in Eagle: ~ 400

Number of Attributes to reflect units in Distributed Eagle:

| | Define | Update |
|---|---|---|
| Ground Movers | 43 | 31 |
| Air Maneuver | 35 | 26 |
| Fixed Wing | 33 | 22 |

Number of Attributes to reflect Aggregate Ground Units in JTFp:

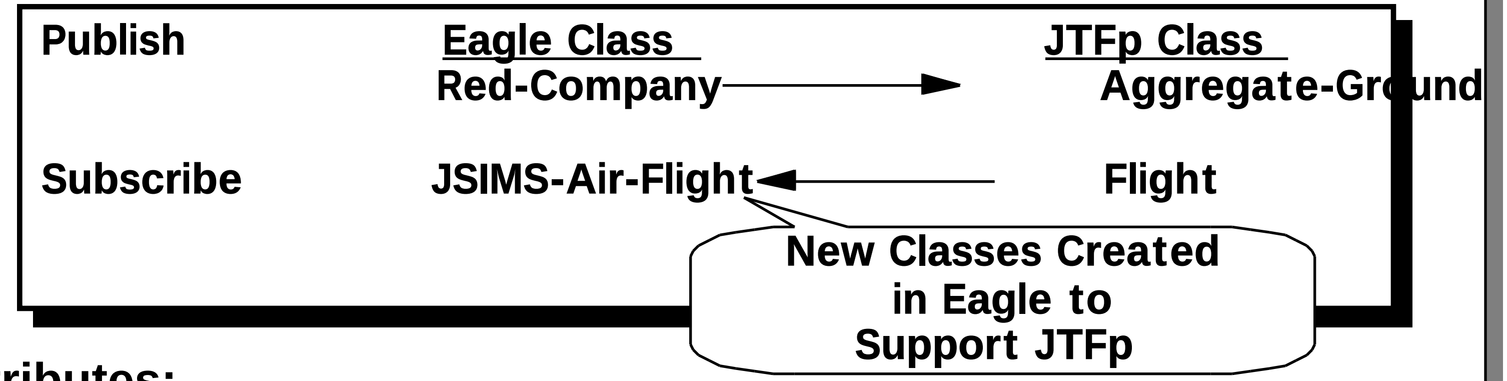
| | Define | Update |
|---|---|---|
| Ground Movers | 29 | 17 |

Attributes used for detection's

# Eagle modifications to support the JTFp Object Class and Attributes

0 Translate the Eagle combat units & attributes to conform to the JTFp FOM and JTFp units & attributes to Eagle.

- Class structure:



Attributes:

| | Eagle ↔ | JTFp |
|---|---|---|
| Attribute Names: | a-side | affiliation |
| Enumerated types: | M1 | 5 |
| Data Translation: | xy (1) | Lat & Lon (2) |
| New Attributes: | <N/A> | Motion Type |
| Methodology: | Time Step Location | Dead Reckoning |
| Data Types | Lisp Symbol | Float etc |

# Contexts
# Eagle & the High Level Architecture

0 **Eagle Design Summary**

0 **Summary of Effort & Design**

0 **Technical Approach**

0 **Class Structures**

❹**Interactions**

0 **Event Synchronization**

0 **RTI Functionality Exercised**

0 **Testing - Results**

0 **Summary**

**Distributed Eagle**
**Shooting**
**Communications**
**Data Base Consistency**
**DIS interactions**
**Surrogate C4I**

**JTFp**
**Federation Management**
**Shooting**
**Communications**
**Environmental**

# Distributed Eagle Interaction Structure

## Typical Interactions between combat units and services

0 **Unit Interactions**

- **Detection (implicit)**
- **Direct Fire**
- **Indirect Fire**
- **Air to ground**
- **Ground to air**
- **Communications**

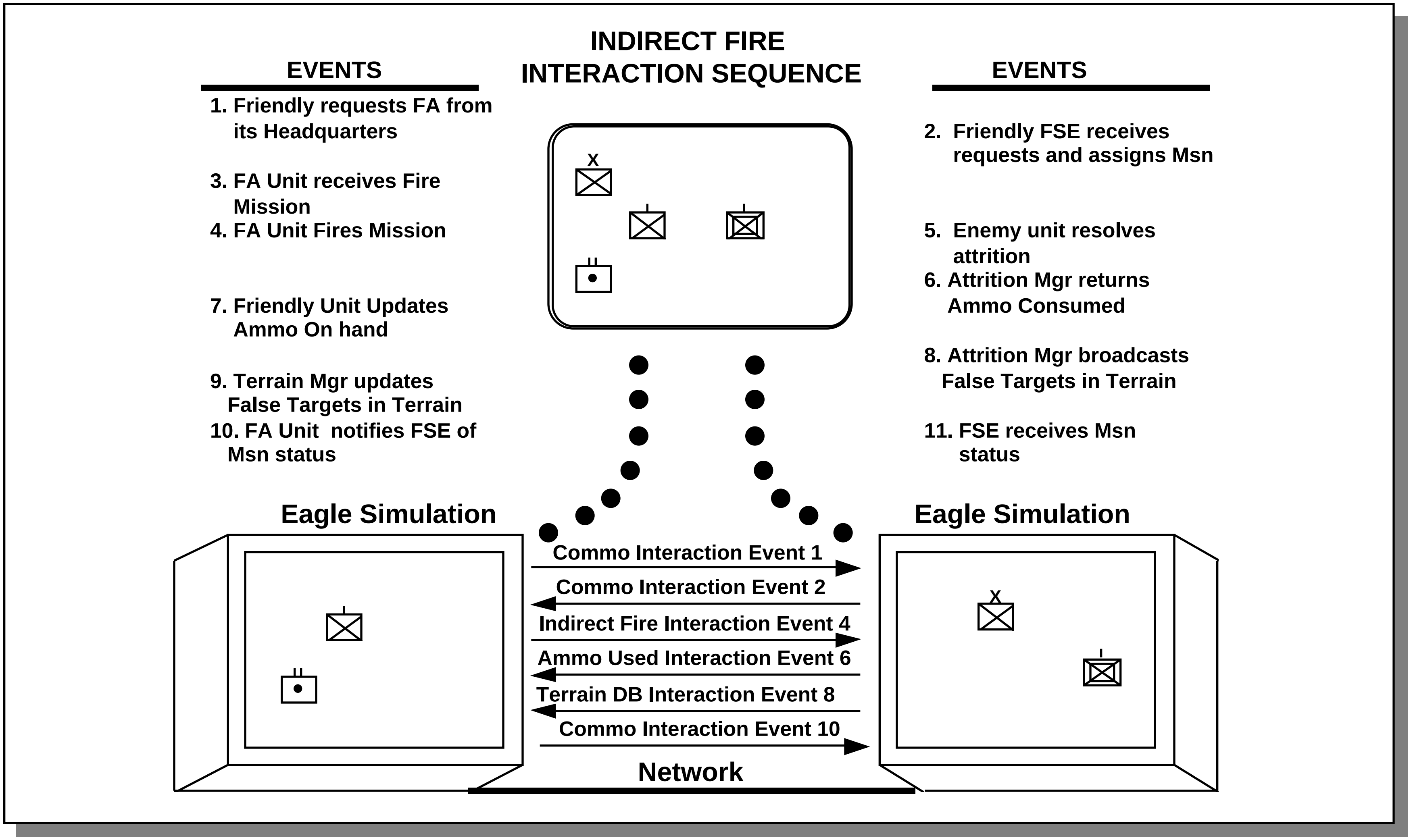
**Command & Control between units is established by the Plan. All reporting and directing go through Commo channels.**

0 **Service Interactions**

- **Event Queue Management**
  - **Add/remove events**
- **Control Measures Database consistency**
  - **Create CM**
- **Terrain Database consistency**
  - **false targets**
  - **register engr work**
- **Attrition**
  - **Ammo use**
  - **Suppression**

# Eagle Interactions - C2 ... Indirect Fire

## Example Interactions in distributed Eagle



INDIRECT FIRE
INTERACTION SEQUENCE

EVENTS

1. Friendly requests FA from its Headquarters

3. FA Unit receives Fire Mission

4. FA Unit Fires Mission

7. Friendly Unit Updates Ammo On hand

9. Terrain Mgr updates False Targets in Terrain

10. FA Unit notifies FSE of Msn status

EVENTS

2. Friendly FSE receives requests and assigns Msn

5. Enemy unit resolves attrition

6. Attrition Mgr returns Ammo Consumed

8. Attrition Mgr broadcasts False Targets in Terrain

11. FSE receives Msn status

X
Eagle Simulation
Eagle Simulation
X
Commo Interaction Event 1
Commo Interaction Event 2
Indirect Fire Interaction Event 4
Ammo Used Interaction Event 6
Terrain DB Interaction Event 8
Commo Interaction Event 10
Network

# Current Interactions Allowed - Distributed Eagle

## 23 interaction types - total with subtypes: 65

| Interaction | Initiating Class | Receiving Class |
|---|---|---|
| Engagement direct fire ground to ground | Attrition Manager | Attrition Manager |
| Bulk ammunition consumed | Attrition Manager | Ground-Movers, Air Movers |
| Engagement indirect fire ground to ground | Attrition Manager | Attrition Manager |
| Suppression update | Attrition Manager | Ground-Movers |
| Engagement Air to Ground | Attrition Manager | Ground-Movers |
| Aircraft abort flight notification. | Fixwing | Airspace manager |
| New air/ada events for Time Step. | Air Space Manager | Air Space Manager |
| Communications between units | Commo Manager | Commo Manager |
| Create control measure - db_consistency | Scenario Control Measures Manager | Scenario Control Measures Manager |
| Dead Target Updates - db_consistency | Attrition Manager | Terrain Manager |
| Register Eng. Work - db_consistency | Terrain Manager | Terrain Manager |
| Update Terrain Feature - db_consistency | Terrain Manager | Terrain Manager |
| Remove Terrain Feature - db_consistency | Terrain Manager | Terrain Manager |
| Ground receive losses air attack. | Attrition manager | Ground Movers |
| Activate IDF msn | Military Unit (Command) | Attrition Manager |
| Update tf occupcany - db_consistency | Terrain Manager | Terrain Manager |
| Update-active-aois | Ground-Mover | Air-Maneuver |
| Create-breach - db_consistency | Terrain Manager | Terrain Manager |
| Create-bypass - db_consistency | Terrain Manager | Terrain Manager |
| Set Unit visibility terrain- db_consistency | Military-units | Terrain Manager |
| Set Unit visibility map- db_consistency | Military-units | Terrain Manager |
| Set unit visible - db_consistency | Military-units | Military-units |
| Db-consistency, used to coordinate events | Aggregate actors | Aggregate actors |
| Eagle Management - stop, start ... | Eagle Controller, Confederation Manager | Eagle Controller, Confederation Manager |
| SIU - Dis interactions between actors | Model Network Manager | Model Network Manager |
| Resolution Unit Interface - C4I interactions | C4I Interface, Military Unit | C4I Interface, Military Unit |
| CommandUnit Interface - C4I interactions | C4I Interface, Military Unit | C4I Interface, Military Unit |

# JTFp Interactions that involve Eagle

0 **Unit Interactions**

- **Detection (implicit)**
- **Indirect Fire**
- **Air to surface**
- **Surface to air**
- **Communications**

0 **Federation Management Interactions**

- **Initialization**

0 **Environmental**

- **LOS**

| Interaction | Initiating | Receiving |
|---|---|---|
| TBM Warming | Player | Eagle Commo Manager - Player |
| TBMLaunchAlert | Player | Eagle Commo Manager- Player |
| Situation Report | Player - Eagle Commo Manager | Player (JTF HQ) |
| RequestAirSupport | Player - Eagle Commo Manager | Player (JTF HQ) |
| AirToAggregate GroundEngage | Player | Eagle Attrition Manager |
| DiscreteGroundToAirEngage | Player | Eagle Attrition Manager |
| AggregateGround ToAirEngage | Eagle Attrition Manager | Player |
| Get & Return LOS | Eagle Terrain Manager | Surface Cover |
| Initialize Federation | Federation Status | Eagle Federation Status |
| Execute Federation | Federation Status | Eagle Federation Status |

**JTFp has defined 24 Interactions**
**Eagle subscribes/published to 10**

# Eagle modifications to support the JTFp Interactions and Parameters

0 **Translate the Eagle interactions & parameters to conform to the JTFp FOM and JTFp interactions & parameters to Eagle.**

- **Interactions:**

| **Publish** | **<u>Eagle</u>** | | **<u>JTFp</u>** |
|---|---|---|---|
| | **engage-ground-to-air** | → | **AggregegateGroundtoA** |
| **Subscribe** | **commo** | ← | **TBM Warning** |
| | **engage-air-to-ground** | ← | **AirtoAggregateGround** |

**Parameters:**

| | **<u>Eagle</u>** | ↔ | **<u>JTFp</u>** |
|---|---|---|---|
| **Parameter Names:** | **a-system** | | **weapon_system** |
| **Enumerated types:** | **M1** | | **5** |
| **Data Translation:** | **xy (1)** | | **Lat & Lon (2)** |
| **New Attributes:** | **<N/A>** | | **Launch_quad** |
| **Data Types** | **Lisp Symbol** | | **Float etc.** |
| **Data Format** | **Complex Type** | | **Separate Parameters** |

# Contexts
# Eagle & the High Level Architecture

0 **Eagle Design Summary**

0 **Summary of Effort & Design**

0 **Technical Approach**

0 **Class Structures**

0 **Interactions**

❹ **Event Synchronization**

0 **RTI Functionality Exercised**

0 **Testing - Results**

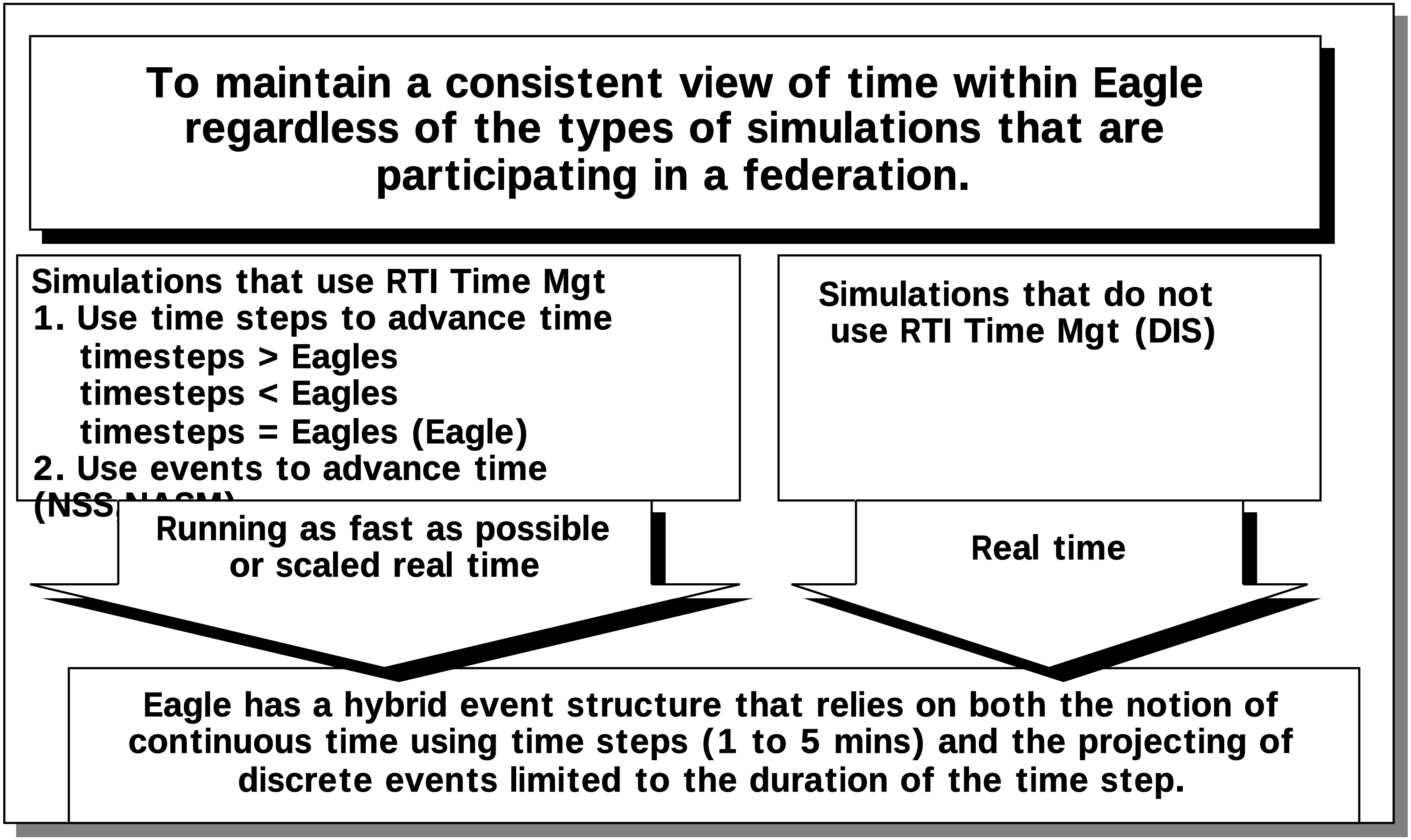
0 **Summary**

**Time Stepped**

**Event**

**Real Time**

# The Challenge



**To maintain a consistent view of time within Eagle regardless of the types of simulations that are participating in a federation.**

**Simulations that use RTI Time Mgt**
**1. Use time steps to advance time**
**timesteps > Eagles**
**timesteps < Eagles**
**timesteps = Eagles (Eagle)**
**2. Use events to advance time (NSS,NASM)**

**Simulations that do not use RTI Time Mgt (DIS)**

**Running as fast as possible or scaled real time**

**Real time**

**Eagle has a hybrid event structure that relies on both the notion of continuous time using time steps (1 to 5 mins) and the projecting of discrete events limited to the duration of the time step.**

# Additional Challenges

1. Eagle can not have unique time management scheme for each Fed.

2. Eagle must maintain causality with DIS. Processing time must be considered in updates.

3. Eagle must maintain consistency between its internal time step events which all occur at the same time.

4. Eagle must constraint time advance to RTI.

5. Eagle must be able to advance to a time short of requested (external events).

**Tools provided by RTI that Eagle uses to maintain time & consistency:**

TIME_ADVANCE_REQUEST

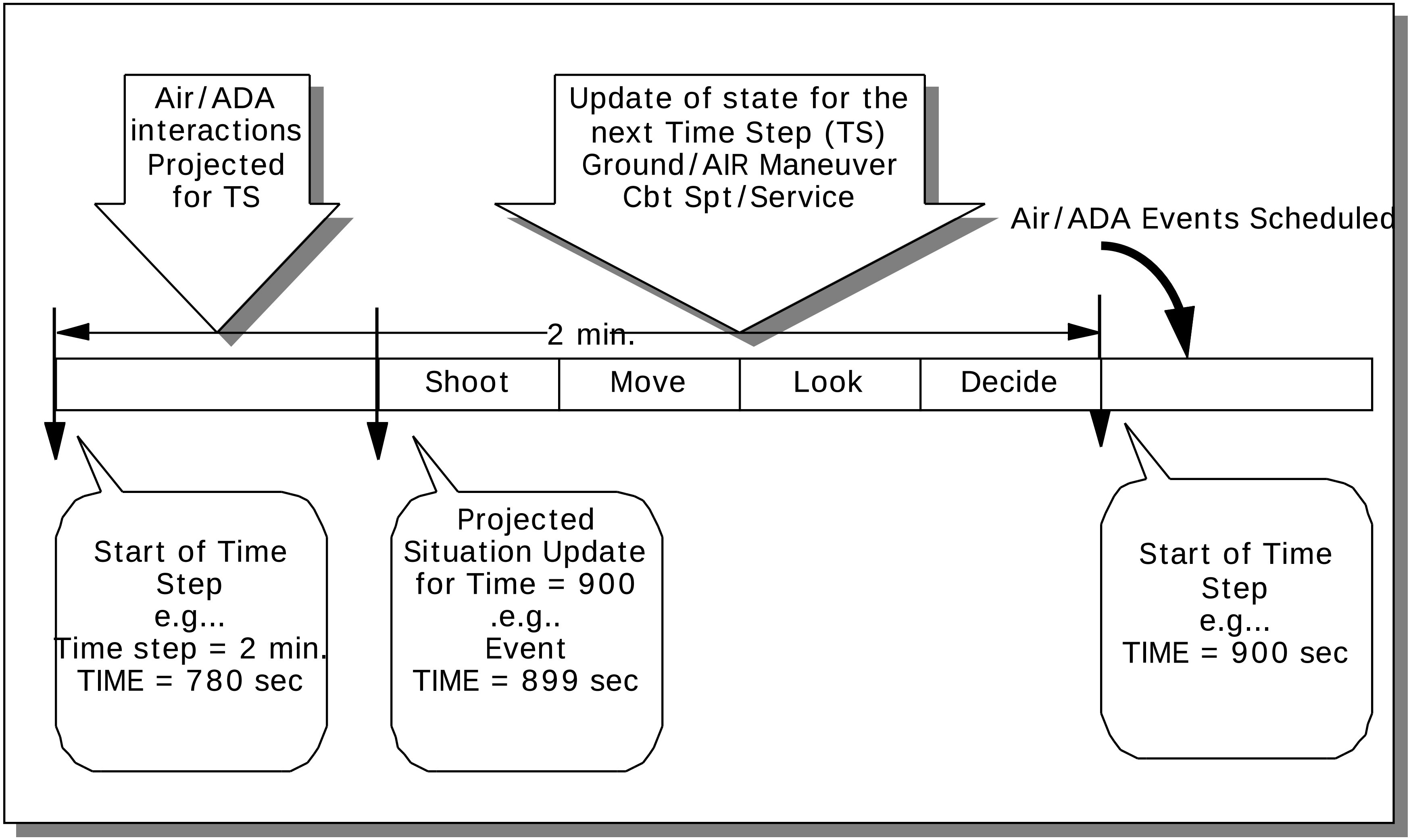
TIME_ADVANCE_GRANT

NEXT_EVENT_REQUEST

SET_LOOKAHEAD

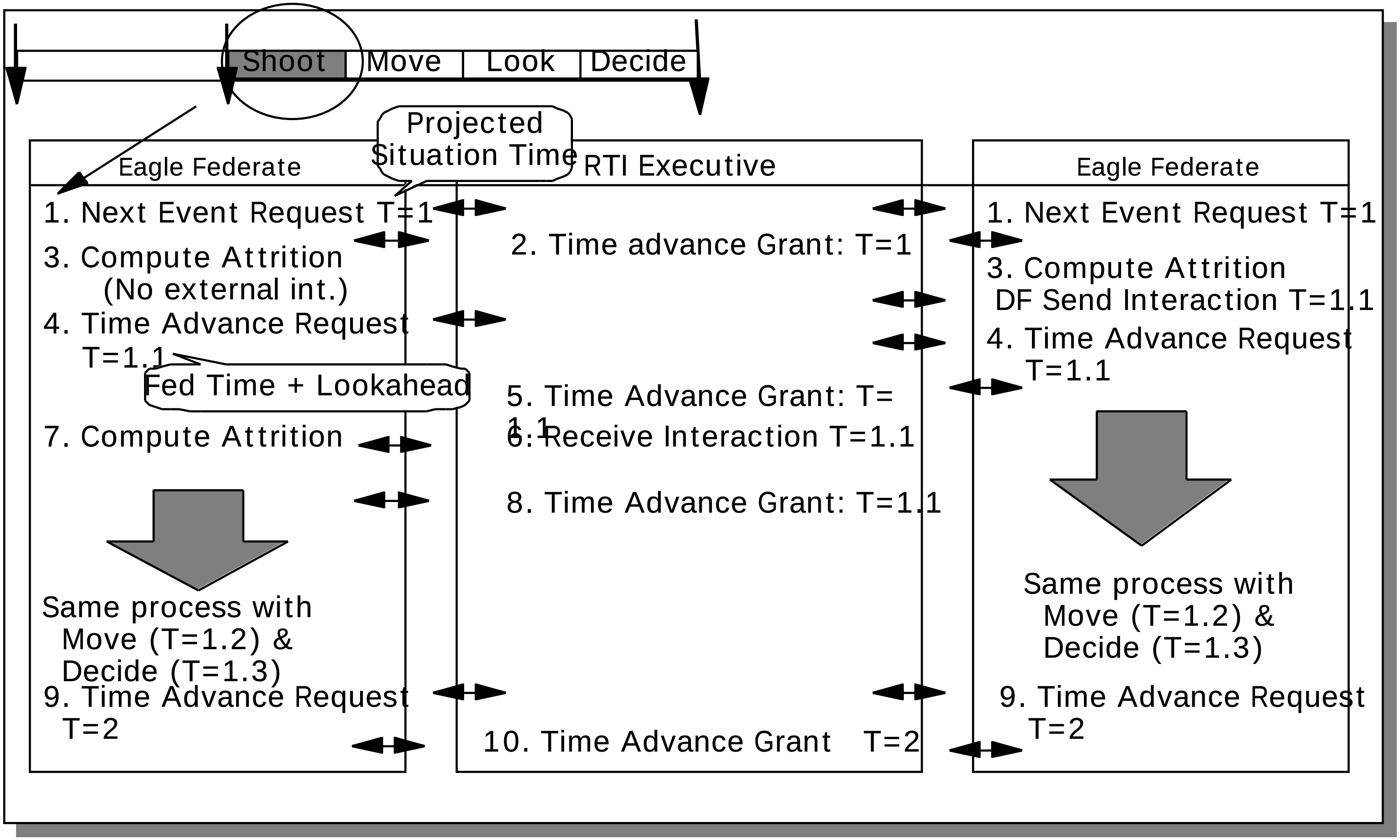
# EAGLE TIME MANAGEMENT

## Hybrid Structure - Time step & Event


Air/ADA interactions Projected for TS
Update of state for the next Time Step (TS) Ground/AIR Maneuver Cbt Spt/Service
Air/ADA Events Scheduled
2 min.
Shoot
Move
Look
Decide
Start of Time Step e.g... Time step = 2 min. TIME = 780 sec
Projected Situation Update for Time = 900 .e.g.. Event TIME = 899 sec
Start of Time Step e.g... TIME = 900 sec

# EAGLE TIME MANAGEMENT

## Time Step Update Process - Projecting Situation for next time Step



Shoot | Move | Look | Decide

Projected Situation Time

Eagle Federate

1. Next Event Request T=1

3. Compute Attrition
(No external int.)

4. Time Advance Request
T=1.1

Fed Time + Lookahead

7. Compute Attrition

Same process with
Move (T=1.2) &
Decide (T=1.3)

9. Time Advance Request
T=2

RTI Executive

2. Time advance Grant: T=1

5. Time Advance Grant: T= 1.1

6. Receive Interaction T=1.1

8. Time Advance Grant: T=1.1

10. Time Advance Grant T=2

Eagle Federate

1. Next Event Request T=1

3. Compute Attrition
DF Send Interaction T=1.1

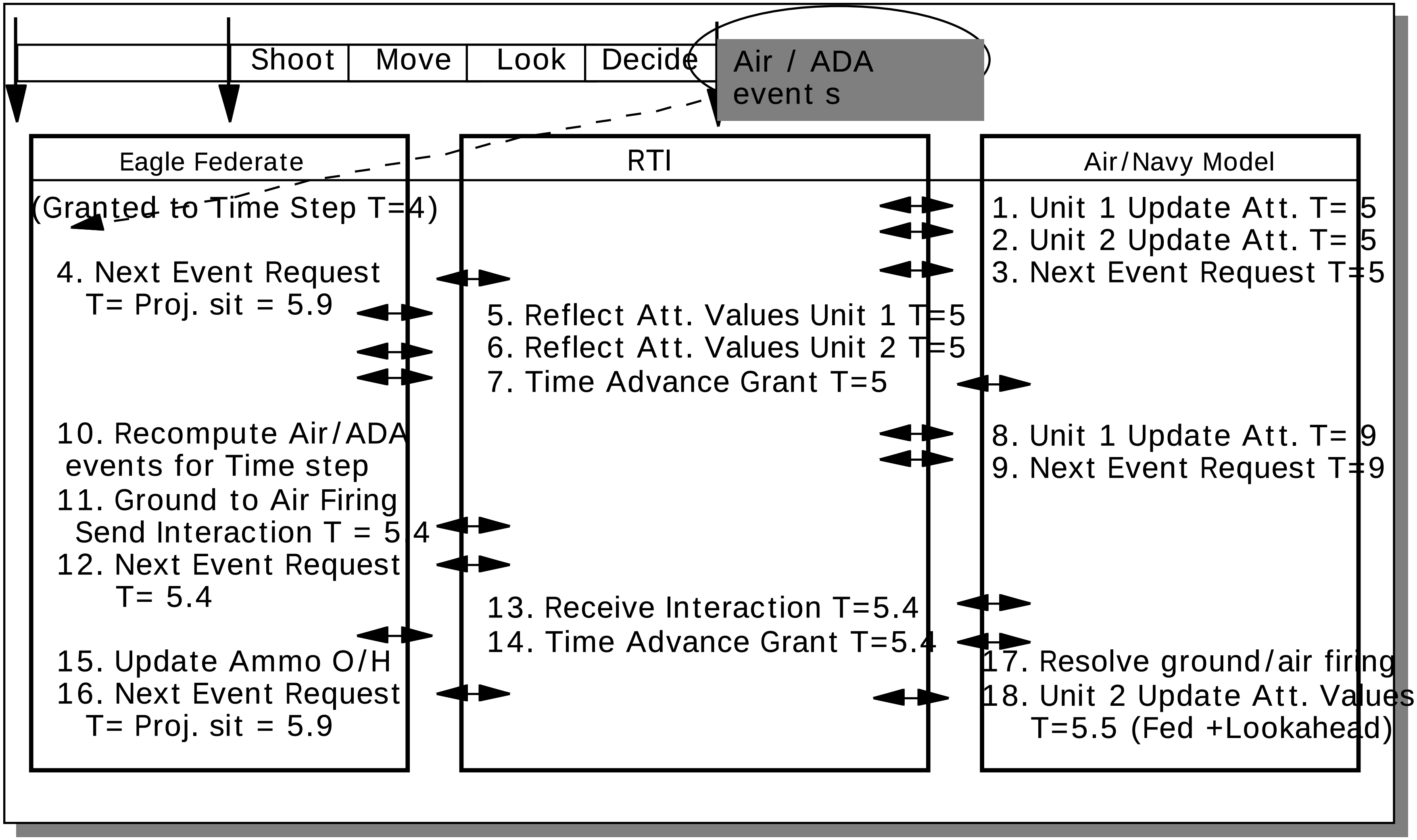
4. Time Advance Request
T=1.1

Same process with
Move (T=1.2) &
Decide (T=1.3)

9. Time Advance Request
T=2

# EAGLE TIME MANAGEMENT

## Event Update Process



Shoot | Move | Look | Decide | Air / ADA event s

Eagle Federate

(Granted to Time Step T=4)

4. Next Event Request
T= Proj. sit = 5.9

10. Recompute Air/ADA
events for Time step

11. Ground to Air Firing
Send Interaction T = 5.4

12. Next Event Request
T= 5.4

15. Update Ammo O/H

16. Next Event Request
T= Proj. sit = 5.9

RTI

5. Reflect Att. Values Unit 1 T=5

6. Reflect Att. Values Unit 2 T=5

7. Time Advance Grant T=5

13. Receive Interaction T=5.4

14. Time Advance Grant T=5.4

Air/Navy Model

1. Unit 1 Update Att. T= 5

2. Unit 2 Update Att. T= 5

3. Next Event Request T=5

8. Unit 1 Update Att. T= 9

9. Next Event Request T=9

17. Resolve ground/air firing

18. Unit 2 Update Att. Values
T=5.5 (Fed +Lookahead)

# Eagle modifications to support the JTFp Event Queue Management

0 **Subordinate  Eagle’s simulation event queue to that of the RTI**

  - **Approval to execute an Eagle event must come from the RTI.**

    =**Request may be denied .**

0 **Ability to respond to new external events from RTI.**

    =**Eagle simulation event queue repair.**

    =**Updating of aircraft situation within Eagle’s timestep.**

    =**Reallocation of ADA to respond to aircraft projected flight paths.**

0 **Maintaining of Federate Time**

  - **Manipulation of time for lookahead requirements.**

# Contexts
# Eagle & the High Level Architecture

0 **Eagle Design Summary**

0 **Summary of Effort & Design**

0 **Technical Approach**

0 **Class Structures**

0 **Interactions**

0 **Event Synchronization**

❹ **RTI Functionality Exercised**

0 **Testing - Results**

0 **Summary**

**Eagle conforms to the Runtime Interfaces**

# RTI - Functional Interfaces Used Distributed & Functional Area

| Type | Avail | Used | Type | Avail | Used |
|---|---|---|---|---|---|
| Federation Management | 17 | 3 | Ownership Management | 7 | 1 |
| Declarararation Management | 6 | 4 | Object Management | 17 | 10 |
| Time Management | 9 | 4 | Data Distribution Management | 6 | 0 |

**Federation Management**

CREATE_FEDERATION_EXECUTION
JOIN_FEDERATION_EXECUTION
RESIGN_FEDERATION_EXECUTION

**Declarararation Management**

PUBLISH_OBJECT_CLASS
SUBSCRIBE_OBJECT_CLASS_ATTRIBUTE
PUBLISH_INTERACTION_CLASS
SUBSCRIBE_INTERACTION_CLASS

**Time Management**

TIME_ADVANCE_REQUEST
NEXT_EVENT_REQUEST
SET_LOOKAHEAD
TIME_ADVANCE_GRANT

**Ownership Management**

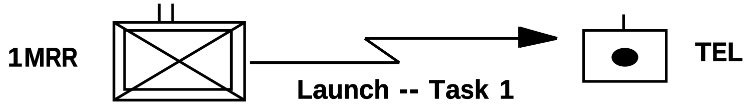
REQUEST_ATTRIBUTE_OWNERSHIP_DIVESTITURE

**Object Management**

ID_REQUEST
REGISTER_OBJECT
DELETE_OBJECT
UPDATE_ATTRIBUTE_VALUES
SEND_INTERACTION
DISCOVERED_OBJECT
REMOVE_OBJECT
REFLECT_ATTRIBUTE_VALUES
RECEIVE_INTERACTION
PROVIDE_ATTRIBUTE_VALUE_UPDATE

**The Above are used & understood by Eagle**

# Functional Interfaces Ownership Management Eagle creates and NASM AP Flies



1. TEL receives msg to Launch Ballistic Missile - Changes to Operational Activity Pre Launch

Create Ballistic Missile - Eagle Class RED-AF-MSL with name “bal_msl_1”

Initialize with Operational Activity SpinUptoLaunch, determine Launch Time

If Func-area-distributed

**Id_request, Register_Object, Update_attribute_values**

If Not Func-area-distributed place launch event on ASM event queue

If Func-area-distributed

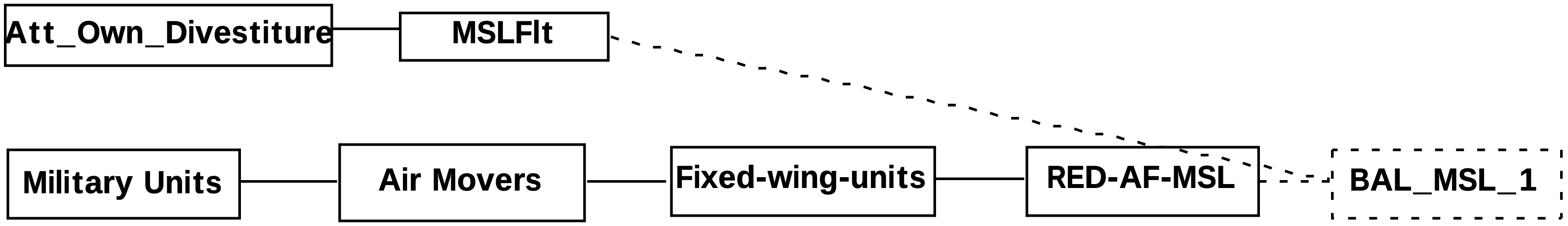
A. Place Divest-Msl-flight event on ASM event queue

B. **Request_attribute_ownership_divestiture**

To NASM - location & Time of arrival attributes

2. Air Space Manger processes Divest-Msl-flight event

Attach Bal_msl_1 to Class MSLFlt. This will cause methods to be inherited by BAL_MSL_1 that will over write normal move & update methods, allowing all movement of the MSL to be caused by updates from the Federation.



# Contexts
# Eagle & the High Level Architecture

0 **Eagle Design Summary**

0 **Summary of Effort & Design**

0 **Technical Approach**

0 **Class Structures**

0 **Interactions**

0 **Event Synchronization**

0 **RTI Functionality Exercised**

❹ **Testing - Results**

0 **Summary**

**Measures of Effectiveness**

**Speed & Consistency**

# Results - Distributed Eagle

## Consistency & Time

**Consistency** - Simple Test, 2 machines

| | | Machine 1 | Machine 2 |
|---|---|---|---|
| Direction | Type | Num | Num |
| OUT | FED | 1 | 2 |
| | DEC | 33 | 33 |
| | OBJ | 355 | 343 |
| | OBJ OUT | 8007 | 7816 |
| | OWN | 0 | 0 |
| | TIME | 723 | 723 |
| | TOTAL | 9119 | 8917 |
| IN | FED | 0 | 0 |
| | DEC | 5 | 5 |
| | OBJ | 7816 | 8007 |
| | OWN | 0 | 0 |
| | TIME | 144 | 144 |
| | TOTAL | 7965 | 8156 |

1. No Messages lost.
2. Duplicates  
   Ver .30 avg: 100  
   Ver .33 none
3. Consistency with standalone  
   ??

IT WORKS

**Time** - 2 machine run 100 units/machine

ALSP : 21 to 1  
HLA (.33) : 24 to 1 (F.0alpha) : 25 to 1

# Contexts
# Eagle & the High Level Architecture

0 **Eagle Design Summary**

0 **Summary of Effort & Design**

0 **Technical Approach**

0 **Class Structures**

0 **Interactions**

0 **Event Synchronization**

0 **RTI Functionality Exercised**

0 **Testing - Results**

❹ **Summary**

# Summary

0 **Code Changes (< 3.0%)**

- **Basic model is app. 750,000 lines of code**
- **Added Translator Interface app. 16,000 lines of Lisp code**
- **Added 3,742 lines of C++ interface code.**

0 **Time**

- **Initial design & coding of Distributed Eagle using ALSP = 10 months**
- **SOM/FOM development = 1 months**
- **Modify the ALSP interface code = 3 months**
- **Creation of C++ interface code = 1 month**
- **Testing = 1 month**

0 **Eagle's Design (Architecture) & Object Oriented approach facilitated transition.**